

**PENERAPAN BASIS DATA CITRA PADA SISTEM Pencarian Citra Berbasis Isi:
MENGUNAKAN FASILITAS *JAVA OBJECT SERIALIZATION* DAN
MENGUNAKAN FASILITAS *MYSQL***

Eliza Margaretha, Eka Aditya, Ade Azurat, Maruli Manurung, dan Aniati Murni

Fakultas Ilmu Komputer, Universitas Indonesia, Depok, Indonesia
elm40@ui.edu, aniati@cs.ui.ac.id

Abstrak

Makalah ini membahas dua pilihan penerapan struktur basis data citra pada sistem pencarian citra berbasis isi. Pendekatan pertama menggunakan *folder* untuk menyimpan berkas citra dan *Java object serialization* untuk menyimpan data citra. Pendekatan kedua menggunakan basis data *Data Base Management System MySQL* untuk menyimpan berkas dan data citra. Kedua pendekatan dibahas dari aspek penerapan struktur basis data untuk tujuan pengembangan sistem pencarian citra berbasis isi yang efisien. Data yang tidak terstruktur dan proses *clustering* data lebih mudah ditangani dengan struktur basis data dari pendekatan pertama. Data yang jumlahnya besar dan terstruktur serta proses *indexing* lebih mudah ditangani dengan struktur basis data dari pendekatan kedua. Sistem pencarian citra berbasis isi lebih banyak melakukan kueri jenis *select* dibandingkan dengan *insert* dan *update* data, dalam hal ini kedua pendekatan dapat memenuhinya dengan baik. Secara umum, pendekatan kedua dianggap memberikan dukungan yang baik dalam penyimpanan dan manipulasi data, serta dapat mengurangi upaya dan waktu yang dibutuhkan pada pengembangan sistem.

Kata kunci : *Java object serialization*, *MySQL*, Sistem pencarian citra berbasis isi

1. Pendahuluan

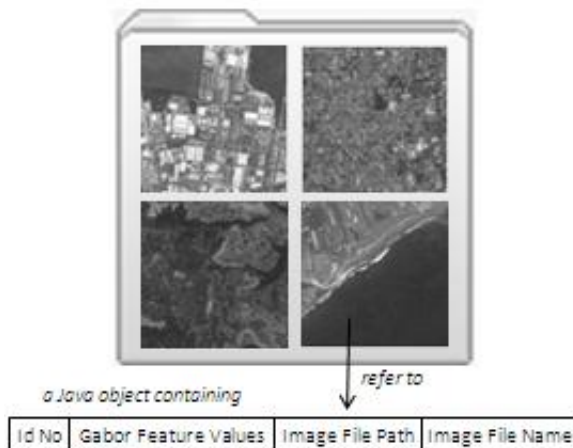
Sejumlah besar sistem pengolahan citra telah dibangun untuk berbagai aplikasi, seperti: aplikasi biomedis dan aplikasi penginderaan jarak jauh. Dalam perkembangannya di masa lalu, teknologi pengolahan citra cenderung berkembang lebih pesat dibandingkan dengan teknologi basis data untuk penyimpanan citra. Data citra umumnya dianggap berukuran besar terutama dibandingkan dengan kemampuan atau kapasitas media penyimpan data yang tersedia, sehingga cara penyimpanan data citra telah menjadi masalah yang penting. Sebagai respon terhadap masalah tersebut, orang berusaha melakukan pemampatan data citra agar dapat disimpan pada media penyimpan yang tersedia. Berbagai teknik pemampatan data yang efisien telah banyak dihasilkan.

Pada masa berkembangnya teknologi basis data dan kemajuan teknologi media penyimpanan, kapasitas media penyimpanan data citra tidak lagi merupakan kendala. Berbagai pendekatan dalam membangun struktur basis data citra banyak bermunculan. Mulai dengan penggunaan *folder*,

representasi dalam bit (*blob*), sampai ke penggunaan *codebook*. Sistem pencarian citra berbasis isi berfungsi untuk mencari citra-citra pada basis data citra yang serupa dengan contoh citra yang diberikan pemakai sistem. Tingkat keserupaan disini dapat didasarkan pada fitur warna, tekstur, maupun bentuk objek yang ada pada citra terkait [1].

Makalah ini membahas dua pendekatan praktis dalam membangun basis data untuk sistem tersebut diatas. Pendekatan yang pertama menggunakan fasilitas *folder* dan *Java object serialization*, sedangkan pendekatan kedua menggunakan fasilitas *MySQL DataBase Management System*. Kedua pendekatan tersebut selanjutnya dibandingkan dengan melihat aspek-aspek yang dibutuhkan pada pembangunan suatu struktur basis data untuk mendapatkan suatu sistem pencarian citra berbasis isi yang efisien. Salah satu aspek antara lain adalah fasilitas *indexing*.

Makalah ini disusun sebagai berikut. Pendekatan pertama dibahas di Bagian 2 dan pendekatan kedua dibahas di Bagian 3. Perbandingan kedua pendekatan dibahas di Bagian 4. Selanjutnya, makalah ini ditutup dengan sebuah ringkasan di Bagian 5.



Gambar 1. Contoh struktur basis data menggunakan folder dan Java object serialization.

2. Konstruksi Basis Data Menggunakan Folder dan Java Object Serialization

Java Object serialization merupakan suatu proses penyimpanan status suatu objek dalam representasi suatu urutan byte, termasuk juga proses pembangunan kembali urutan byte tersebut menjadi objek terkait di masa yang akan datang ketika objek tersebut dibutuhkan [2]. Java serialization API (Application Programming Interface) menyediakan mekanisme standar untuk pengembang dalam mengelola proses object serialization tersebut.

Dalam membangun suatu sistem pencarian citra berbasis isi untuk aplikasi penginderaan jarak jauh, Aditya dkk. [3] telah menggunakan sistem basis data yang sederhana. Citra disimpan pada suatu folder yang telah ditentukan, dan fasilitas Java object serialization digunakan untuk merekam informasi citra terkait. Struktur basis data dengan pendekatan ini dapat dilihat pada Gambar 1.

Citra-citra foto satelit disimpan pada suatu folder. Pada contoh aplikasi tersebut, untuk setiap citra satelit dikaitkan dengan informasi yang disimpan sebagai rekaman yang terdiri dari nomor identitas, atribut isi citra yang direpresentasikan dalam nilai-nilai fitur Gabor, kelas-kelas objek penutup lahan yang ada pada citra terkait disertai dengan informasi luasannya, petunjuk path dimana berkas citra terkait disimpan, serta nama berkas citranya. Informasi ini disimpan sebagai objek dari suatu Java class yang menerapkan antar muka java.io.Serializable Interface. Gambar 2 menyajikan contoh dari kode suatu Java class yang mengandung informasi nama berkas citra yang akan disambungkan atau dikaitkan.

Selanjutnya, penetapan objek terkait juga perlu dilakukan dengan menggunakan fasilitas java.io.ObjectOutputStream class. Pada akhirnya,

proses penyambungan diselesaikan dengan menuliskan objek tersebut pada suatu berkas yang spesifik. Gambar 3 menyajikan contoh kode yang diperlukan untuk melaksanakan prosedur tersebut. Objek yang telah ditetapkan tersebut dapat dibangun kembali dengan membaca urutan byte yang tersimpan pada berkas terkait, dan objek dapat dibangun kembali sebagai replika dari objek aslinya.

```
import java.io.Serializable;

public class ImageRecord implements
Serializable {
    public void setImageName(String imageName){
        this.imageName = imageName;
    }
    public String getImageName(){
        return this.imageName;
    }
}
```

Gambar 2. Contoh kode suatu Java class yang menerapkan proses java.io.Serializable Interface.

```
import java.io.ObjectOutputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public static void main(String[] args){
    String filename = "object.ser";
    ImageRecord object = new ImageRecord();
    try {
        OutputStream outputStream = new
            FileOutputStream(filename);

        ObjectOutputStream out = new
            ObjectOutputStream(outputStream);
        out.writeObject(object);
        out.close();
    }
    catch(IOException e){
        e.printStackTrace();
    }
}
```

Gambar 3. Contoh kode suatu Java main class untuk penetapan suatu objek.

Sistem pencarian citra berbasis isi tersebut menggunakan proses kueri berdasarkan suatu contoh citra. Sistem akan memberikan respon dalam bentuk sederetan citra yang serupa dengan contoh citra yang diberikan pengguna, dengan urutan mulai dari citra yang paling serupa sampai ke citra yang paling tidak serupa [3]. Proses dimulai dengan melakukan ekstraksi nilai-nilai fitur Gabor dari citra contoh. Nilai-nilai fitur Gabor yang diperoleh kemudian dibandingkan dengan nilai-nilai fitur Gabor dari setiap citra yang ada di basis data. Tingkat keserupaan antara citra contoh dan setiap citra pada basis data diukur dengan metrik jarak Euclidean. Pada akhirnya sistem menampilkan citra-citra yang mempunyai keserupaan dengan contoh citra, dimulai

dengan citra yang memiliki jarak Euclidean terkecil dengan citra contoh.

3. Konstruksi Basis Data Menggunakan MySQL

MySQL adalah *open source SQL (Structured Query Language) DBMS* yang dikembangkan, didistribusikan, dan didukung oleh MySQL AB, suatu perusahaan komersial yang didirikan oleh pengembang-pengembang MySQL [4]. MySQL merupakan suatu *server* yang melakukan proses penambahan, penggunaan, serta pemrosesan data yang disimpan suatu basis data pada komputer.

Dalam membangun basis data sistem pencarian citra berbasis isi untuk aplikasi batik, E. Margaretha telah menggunakan MySQL. Gambar 4 di bawah ini menyajikan struktur basis data dari sistem yang dibangun.

Field	Type
ImageID	integer
ImageName	varchar
ImageFile	longblob
ImageOrigin	varchar
ImageCategory	varchar
GaborValue	longblob

Gambar 4. Struktur basis data menggunakan MySQL untuk sistem pencarian citra batik berbasis isi.

Pada pengembangan awal, basis data yang dibangun hanya terdiri dari satu tabel. Isi dari tabel adalah sebagai berikut:

- ImageID menyatakan nomor identitas citra batik yang direpresentasikan dalam *integer*
- ImageName menyatakan nama berkas citra batik yang direpresentasikan *varchar*
- ImageFile merupakan berkas citra dalam bentuk *blob*
- ImageOrigin menyatakan tempat asal dari pola batik yang direpresentasikan dalam *varchar*
- ImageCategory menyatakan nama dari kategori atau jenis atau tipe atau pola batik yang dinyatakan dalam *varchar*
- GaborValue menyimpan nilai-nilai fitur Gabor yang dihitung dari citra batik dan dinyatakan dalam *real*

Pada awalnya perlu dibangun basis data dalam MySQL sebelum tabel basis data dibuat. Proses pembentukan basis data dalam MySQL dilakukan dengan mengeksekusi suatu kueri dari Java. Dalam

```
// Specify the MySQL server address
// (localhost) and the database name
// (multilab)
String URL =
"jdbc:mysql://localhost/multilab";

// Specify the username and password to log
// in to MySQL Server
String Username = "root";
String Password = "";

try {
    // Load Java driver for MySQL
    Class.forName("com.mysql.jdbc.Driver");
    Connection connection =
        DriverManager.getConnection(URL, Username,
            Password);
}
catch(Exception e){}
```

Gambar 5. Contoh kode Java untuk membangun koneksi antara Java dan MySQL.

```
String filename = "kawung.jpg";
PreparedStatement pstmt = null;

// Specify table name (batik) and field name
// (ImageName)
pstmt = connection.prepareStatement("INSERT
    INTO batik(ImageName) VALUES(?)");

// Specify the value
pstmt = setString(1, filename);
pstmt = executeUpdate;
```

(a)

```
Statement stmt =
    connection.createStatement();
ResultSet rs = stmt.executeQuery("SELECT
    ImageName FROM batik");
```

(b)

Gambar 6. Contoh kode Java untuk menjalankan perintah kueri SQL menggunakan:
(a) *java.sql.PreparedStatement*; dan
(b) *java.sql.Statement Interfaces*.

hal ini perlu dibangun koneksi antara Java dan MySQL. Suatu Java driver telah digunakan untuk mengubah JDBC (Java Database Connectivity) menjadi koneksi ke protokol jaringan yang digunakan oleh MySQL. Gambar 5 memperlihatkan contoh kode untuk membangun koneksi tersebut.

Setelah koneksi dibangun, maka basis data dengan MySQL dapat diakses dengan menggunakan beberapa *classes* yang ada di *java.sql package*. Pada dasarnya, ada dua cara untuk menjalankan perintah kueri pada SQL. Cara yang pertama dapat menggunakan instruksi *interface java.sql.Statement* untuk melaksanakan perintah kueri SQL yang statis. Sebagai contoh, instruksi tersebut dapat digunakan untuk menjalankan kueri SQL *select*. Cara yang kedua dapat menggunakan *interface*

java.sql.PreparedStatement untuk melaksanakan perintah kueri *SQL* yang dinamis dan biasanya memerlukan konversi parameter. Sebagai contoh, instruksi tersebut dapat digunakan untuk menjalankan kueri *SQL update*.

Pada pengembangan aplikasi batik, *java.sql.PreparedStatement* interface telah digunakan untuk menyusun tabel dari basis data sistem pencarian citra batik berbasis isi. Selain itu, untuk memilih data dari tabel, telah digunakan *java.sql.Statement* interface. Gambar 6 memperlihatkan cara penggunaan kedua *interfaces*.

Sistem yang dibangun menjalankan proses pemilihan data dari tabel basis data menggunakan *PHP*. Koneksi antara *PHP* dan *MySQL* dilakukan dengan *ADODB* yang merupakan suatu *database abstraction library* untuk *PHP* yang digunakan untuk melakukan koneksi dengan *server* basis data. Selanjutnya, kueri *SQL select* dapat dijalankan melalui koneksi yang sudah diciptakan oleh *ADODB*.

Gambar 7(a) menyajikan contoh kode *PHP* untuk membangun koneksi antara *PHP* dan *MySQL*, dan Gambar 7(b) memperlihatkan contoh kode untuk menjalankan kueri *SQL select*.

4. Pembahasan

Pembahasan mencakup beberapa aspek dari basis data yang dapat mendukung efisiensi dari suatu sistem pencarian citra berbasis isi. Selain itu juga akan dibandingkan penyusunan struktur basis data menggunakan dua pendekatan yang dibahas pada Bagian 2 dan 3.

Suatu sistem pencarian data cenderung membutuhkan respon yang cepat untuk setiap kueri *SQL select*. Untuk itu fasilitas *data indexing* menjadi penting untuk mencapai optimasi dari pelaksanaan proses kueri. Proses *indexing* digunakan untuk mengelola struktur internal basis data, terutama dengan mengenali bagian data yang sering diakses sehingga kueri bagian data tersebut dapat dilaksanakan dengan efisien. Dengan kata lain, dengan *indexing* pencocokan kueri dapat dilakukan secara efektif dan proses pencarian pada basis data dapat dilaksanakan dengan lebih efisien.

Struktur basis data yang dibangun dengan pendekatan *folder* dan *Java object serialization*, memungkinkan pengembang untuk membangun sistem *indexing* yang sesuai kebutuhan. Sebagai contoh: pada aplikasi dokumen teks dapat dibuat indeks berdasarkan kata kunci, dan pada aplikasi citra dapat dibuat indeks berdasarkan blok kunci (*key block*) [5]. Walaupun untuk merancang dan menerapkan *indexing* pada struktur basis data ini memerlukan upaya tertentu, namun struktur basis

```
// Setting ADOdb directory
global $adodb_dir;
$adodb_dir = dirname(__FILE__)."/adodb/";
```

```
// Setting database type
global $database_used;
$database_used = "mysql";
```

```
// Setting connection option
global $persistent_connection;
$persistent_connection = true;
```

```
// Setting database account
global $db_username;
global $db_password;
global $db_server;
global $db_database_name;
$db_username = "root";
$db_password = "";
$db_server = "localhost";
$db_database_name = "multilab";
```

(a)

```
// Specify table name (batik) and field name
// (ImageName)
$query = "select ImageName from batik";
$result = $this->execute($query);
$rows = $result->GetRows();
```

(b)

Gambar 7. Contoh kode *PHP* untuk: (a) membangun koneksi antara *PHP* dan *MySQL*; dan (b) menjalankan kueri *SQL select*.

data ini sangat berguna untuk menyimpan dan menyusun indeks dari citra yang isinya tidak terstruktur yang direpresentasikan dalam bentuk graf [5,6].

Pada sisi yang lain, struktur basis data yang menggunakan *MySQL* secara mudah dapat langsung menggunakan prosedur yang tersedia untuk *indexing*. Sistem pengelolaan basis data (*DBMS*) seperti *MySQL* telah dianggap sebagai sistem yang tangguh untuk melaksanakan penyimpanan dan perancangan *indexing* data yang terstruktur. Selain itu, struktur basis data ini secara praktis dapat menangani jumlah data yang besar. *Indexing* pada *MySQL* dapat dilakukan dengan sangat mudah, hanya mengatur beberapa variabel. Bagian data yang sering diakses dengan mudah dipilih dan diindeks. *MySQL* memberikan fleksibilitas untuk mengubah struktur basis data dan menyediakan mekanisme *indexing* yang kompleks. Walaupun struktur basis data ini membutuhkan waktu perancangan yang lebih lama, namun tetap menjamin waktu dan usaha pengembangan yang relatif lebih sedikit.

Bagaimanapun, efisiensi dari perancangan *indexing* sangat dipengaruhi oleh karakteristik dari data. Data yang mempunyai struktur yang kompleks terkadang lebih baik dicari menggunakan representasi struktur data seperti *Heap Tree* dari pada struktur data

B-Tree. Dalam prakteknya, pengguna sangat tergantung pada fasilitas struktur data dan prosedur *indexing* yang disediakan. Selain itu, *DBMS* juga tidak biasa digunakan untuk membangun struktur basis data yang datanya tidak terstruktur dengan baik, dimana transformasi data ke bentuk tabel-tabel yang berhubungan menjadi sulit.

DBMS juga menyediakan dukungan untuk mengatur wewenang untuk akses data terutama dalam konteks pengguna yang jamak, penggunaan berbagai antar muka untuk pengguna yang jamak, serta pengelolaan integritas data [7]. Dukungan tersebut tidak tersedia dan harus diterapkan secara manual pada struktur basis data dengan pendekatan pertama (*folder* dan *Java object serialization*). Karena sistem pencarian citra berbasis isi lebih banyak menggunakan kueri berdasarkan *SQL select* dibandingkan dengan kueri *SQL insert* dan *SQL update*, maka aspek basis data yang berkaitan dengan proses penyisipan dan proses pembaruan data, seperti pengelolaan akses secara konkuren dan penanganan redundansi data, menjadi tidak terlalu penting.

Optimasi kueri juga dapat ditingkatkan melalui proses *clustering* data yang sering diakses. Proses *clustering* juga dapat mengurangi jumlah data yang harus dibandingkan dengan contoh citra pada suatu kueri. Secara spesifik dapat dikatakan bahwa contoh citra hanya perlu dibandingkan dengan kelompok data (*cluster*) yang mempunyai nilai atribut yang serupa. Mekanisme proses *clustering* dapat diterapkan pada struktur basis data dengan pendekatan pertama dengan cara yang sederhana, yaitu dengan membangun *subfolder* untuk setiap kelompok data dan menyusun program *clustering*. Di sisi lain, pada struktur basis data dengan pendekatan kedua, dapat diupayakan pembuatan tabel-tabel atau atribut baru yang mendefinisikan adanya kelompok (*cluster*) pada data.

5. Penutup

Makalah ini membahas dua struktur basis data. Model yang pertama menggunakan *folder* dan *Java object serialization*, dan model kedua menggunakan basis data *MySQL*. Fokus pembahasan adalah pada aspek struktur basis data yang mendukung terpenuhinya kebutuhan sistem pencarian citra berbasis isi yang efektif dan efisien.

Beberapa karakteristik yang sering dijumpai pada sistem pencarian citra berbasis isi dan proses-proses yang sering dibutuhkan antara lain adalah:

- 1) Proses *indexing* dan *clustering* merupakan proses yang dibutuhkan untuk mendapatkan sistem yang efisien;

- 2) Lebih banyak melakukan kueri proses pencarian (*select*) daripada penyisipan (*insert*) dan pembaruan (*update*) data;
- 3) Sering dijumpai adanya kebutuhan membangun basis data citra yang tidak terstruktur; dan
- 4) Jumlah data yang disimpan juga cukup besar.

Dari pembahasan yang telah dirinci di Bagian 4, dapat disimpulkan bahwa:

- 1) Untuk data yang terstruktur, *DBMS MySQL* (model kedua) menyediakan kemudahan dalam proses *indexing* dibandingkan dengan model pertama.
- 2) Proses *clustering* dengan mudah diterapkan pada model pertama, yaitu dengan merancang *sub-folder* dan program *clustering*. Pada model kedua perlu disusun tabel atau atribut baru yang memberikan informasi tentang adanya *cluster* pada data.
- 3) Data yang tidak terstruktur memerlukan struktur data yang khusus yang biasanya bukan termasuk pada struktur data generik yang disediakan di *DBMS MySQL* (model kedua). Dalam hal ini model pertama lebih cocok untuk menangannya.
- 4) Sistem pencarian citra berbasis isi lebih banyak menggunakan kueri *select* dibandingkan kueri *insert* dan *update* data. Dalam hal ini kedua model mendukung kebutuhan tersebut dengan baik.
- 5) Untuk data yang berukuran sangat besar, model kedua dapat mendukung pengelolaan data secara lebih mudah dan fleksibel.

Sekalipun model *DBMS MySQL* tidak begitu cocok untuk data yang tidak terstruktur, pada umumnya model ini dianggap cukup baik untuk menangani jumlah data yang besar, dan memberikan dukungan yang baik dalam penyimpanan dan manipulasi data, serta dapat mengurangi upaya dan waktu yang dibutuhkan pada pengembangan sistem.

6. Penghargaan

Rasa terima kasih ditujukan kepada Bapak Andreas Febrian yang telah membantu dalam penerapan sistem dan Bapak Ahmad Nizar yang telah memberikan saran-saran yang berharga.

REFERENSI

- [1] T. Andrysiak and M. Choras, "Image Retrieval Based On Hierarchical Gabor Filters," *Int. J. Appl. Math. Comput. Sci.*, vol. 15, no. 4, pp. 471–480, 2005.
- [2] T. Greanier. "Discover the secrets of the Java Serialization API," Sun Developer Network: <http://java.sun.com/developer/technicalArticles/Programming/serialization/>, July, 2000 [September 12, 2008].
- [3] E. Aditya, A. Murni, and Wiweka. "Information Mining in Remote Sensing Application," The 1st Indonesian Geospatial Technology Exhibition: Poster Presentation, Jakarta, August 23-27, 2006.
- [4] MySQL AB. "MySQL 3.23, 4.0, 4.1 Reference Manual," 1997-2008.
- [5] E. G. M. Petrakis, C. Faloutsos, and K. -I. Lin. "ImageMap: An Image Indexing Method Based on Spatial Similarity," *Knowledge and Data Engineering, IEEE Transactions*, Volume 14, Issue 5, pp.979-987, September/October 2002.
- [6] S. Aksoy. "Modeling of Remote Sensing Image Content using Attributed Relational Graphs," Department of Computer Engineering, Bilkent University, Ankara, 06800, saksoy@cs.bilkent.edu.tr, [March 1, 2008].
- [7] R. Elmasri and S. B. Navathe. *Fundamental of Database Systems*, Pearson Education, Inc, MA: Boston, 2004.